EQUISPACED PARETO FRONT CONSTRUCTION FOR CONSTRAINED BIOBJECTIVE OPTIMIZATION*

VICTOR PEREYRA†, MICHAEL SAUNDERS‡, AND JOSE CASTILLO§

Abstract. We consider constrained biobjective optimization problems. One of the extant issues in this area is that of uniform sampling of the Pareto front. We utilize equispacing constraints on the vector of objective values, as discussed in a previous paper dealing with the unconstrained problem.

We present a direct and a dual formulation based on arc-length homotopy continuation and illustrate the direct method (using standard nonlinear programming tools) on some problems from the literature. We contrast the performance of our method with the results of three other algorithms, showing several orders of magnitude speed-up with respect to evolutionary algorithms, while simultaneously providing perfectly sampled fronts by construction. We then consider a large-scale application: the variational approach to mesh generation for partial differential equations in complex domains. Balancing multiple criteria leads to significantly improved mesh design.

Key words. multiobjective optimization, Pareto fronts, constrained optimization, mesh design, parallel global method

AMS subject classifications. 65N50, 65K10

1. Introduction.

Constrained multiobjective optimization arises frequently in engineering and scientific applications. The key optimality concept is that of a *local Pareto optimal point*, which is a point such that no improvement in all the objectives can be achieved by moving to a neighboring feasible point. One of the issues that makes these problems quite different from single-objective nonlinear programming is that the set of Pareto optimal points is usually a continuum that may have disjoint components.

In general, for a problem with p objectives, the Pareto set is a manifold of dimension $p-1$. For instance, in the simplest biobjective case, the Pareto set will be a curve (or collection of curve segments).

An important tool, the *Pareto front*, is the image of the Pareto set viewed in objective space. Again in the biobjective case, that we consider in detail in this paper, this graph presents in a compact way all that one needs to choose an appropriate Pareto-point compromise between the usually conflicting objectives.

Unless there is an *a priori* set of rules to drive an algorithm toward a desired Pareto point, one would want to have a full discrete approximation of the Pareto front in order to make an informed decision. It is now possible to calculate a discrete approximation to the Pareto front for significant problems because of the availability of high performance computers.

Some popular methods for mapping the Pareto front are based on Genetic Algorithms (GA) [4], continuation or homotopy [21, 22, 25], and an interesting geometrically motivated method [3]. In most of these papers a fundamental issue is how to produce economically a well sampled Pareto front.

*Version of March 2, 2010. Technical Report SOL 2010-1.

†Weidlinger Associates Inc., 399 W. El Camino Real #200, Mountain View, CA 94040 (vpereyra@yahoo.com). Partially supported by National Science Foundation SBIR Phase IIB grant DMI-0321420.,

‡Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, CA 94305-4026 (saunders@stanford.edu). Partially supported by Office of Naval Research grant N00014-08-1-0191.

§Computational Science Research Center, San Diego State University, 5500 Campanile Drive, San Diego CA 92182-7720 (castillo@myth.sdsu.edu).

In a previous paper [20], one of the authors has introduced a new idea into marching and global homotopy methods for unconstrained biobjective problems that produces an equispaced discrete representation of the Pareto front using few function evaluations, as compared with genetic algorithms. The approach used there consisted of implementations of Newton's method to solve the first-order optimality conditions. Starting from an optimal point of the individual objectives (themselves Pareto optimal points) and using arc-length homotopy continuation [16, 13], the method solves a sequence of constrained optimization problems involving the problem variables, the homotopy variable λ, and an added nonlinear constraint that requires the vector of objective values to be a given distance from the previous one. *This is the key to producing an equally spaced Pareto front.*

Here we extend the approach to the fully constrained case. The main difference is that we use SNOPT [11], an established nonlinear programming code, to solve the successive single-objective constrained problems that arise from taking convex combinations of the objectives and performing homotopy continuation. In this case, including the additional variable λ and the equispacing constraint into SNOPT is straightforward. Although scalarization by convex combinations has some limitations, as pointed out in the literature, it is one of the most common and natural approaches used to generate Pareto points.

The difficulty until now has been that it is not obvious how to choose the values of λ in order to generate an equispaced front, since the map from weights to optimal Pareto points is nonlinear, and then the mapping from those points to objective space adds another layer of nonlinearity. That is the question that the equispacing constraint answers efficiently, as we demonstrate on a collection of examples from the literature and in a real application to the generation of optimal, well behaved meshes in two-dimensional complex regions for the solution of partial differential equations. We also get a new insight on the intrinsic parametrization of the Pareto front by arc length (in biobjective problems) that is obtained with the addition of the equispacing constraint.

We use an implementation of the direct algorithm to compare it with some problems from the literature [4, 3, 14] and apply it to the mesh generation problem on several regions. We show then the correlation between the mesh quality and the errors obtained when solving an elliptic equation using those meshes.

Variational methods to generate "good" conforming meshes on curved regions have a long history (see for instance Thomson [26]). These methods have shown some difficulties for complicated regions, producing folded, crimped or rough meshes. Castillo [2] has demonstrated that combining linearly several criteria produces better behaved meshes than any single criterion alone.

We consider the generation of conforming, logically rectangular meshes in 2D curved regions using two criteria: minimization of the sum of the squares of all the cell sides and minimization of the sum of squares of all the cell areas. We attack this problem as a biobjective optimization via homotopy continuation.

Although global methods tend to be more complex than simple marching homotopy continuation, especially for large problems, they may be promising in terms of parallel implementation. As far as we know, nobody has explored that possibility yet. We therefore describe in section 6 a parallel algorithm for the global solution of the equispaced Pareto front representation that could be applied to large-scale problems with expensive goals and many constraints and/or variables.

1.1. Related work. Recently we became aware of some additional pertinent work. In [15], Leyffer presents a global method and elegant theory for calculating an approximately equispaced discrete representation of the Pareto front (where "global" means that all points are calculated at once, as in the evolutionary algorithms or the algorithm in Pereyra [20]). Leyffer uses a bilevel optimization approach (different from [20]), with a measure of quality of the discrete representation due to Sayin [24]. The second-level subproblems can be treated as mathematical programs with complementarity constraints (MPCCs), allowing the use of general-purpose nonlinear solvers on the subproblems: an aim that we share. A difficulty is that the subproblems are nonconvex, and the MPCC constraints involve the first derivatives of the goal functions, so that the general-purpose subproblem solvers will typically need higher derivatives.

More to the point of this paper, G. Eichfelder in her PhD thesis [5] and in a recently published book [7] proposes a continuation method that is similar to the homotopy method advocated here. The main differences are that she uses a different scalarization approach due to Pascoletti and Serafini [17], and instead of the equidistance constraint that we propose, she uses a local approximation of the efficient set based on the Lagrange multipliers to estimate an appropriate continuation parameter. There may be a difficulty with this approach if the distance between the points is large, but the methods are otherwise equivalent in complexity [8]. The additional constraint in our case will be negligible in real large-scale problems with expensive goals and many constraints. On the other hand, the work of Eichfelder is more general, adds considerable theoretical background, an extensive set of numerical tests, and a very impressive real application to radiology. Some further references to her work are [6] and some forthcoming papers [9, 10]. It would be possible to combine the two approaches using Eichfelder's method to predict the next point in the continuation even for large steps and then using our approach to correct it, if necessary, to attain equispacing of the discrete Pareto front.

1.2. Notation. The constrained multiobjective optimization problem is defined as

$$\min_{\mathbf{x}} \ \mathbf{f}(\mathbf{x}) \quad \text{subject to} \quad \mathbf{g}(\mathbf{x}) \leq 0, \quad \mathbf{x} \in D, \tag{1.1}$$

where $\mathbf{x} \in R^n$, $\mathbf{f} \in R^p$, $\mathbf{g} \in R^m$, D is a convex set, and the inequalities are applied componentwise. We assume that the nonlinear functions in $\mathbf{f}$ and $\mathbf{g}$ are smooth. For each separate objective function in $\mathbf{f}(\mathbf{x})$, we define $\mathbf{x}_i^*$, $i = 1, \ldots, p$ to be a local minimizer of the associated single-objective problem:

$$\mathbf{x}_i^* \in \arg\min_{\mathbf{x}} \ f_i(\mathbf{x}) \quad \text{subject to} \quad \mathbf{g}(\mathbf{x}) \leq 0, \quad \mathbf{x} \in D. \tag{1.2}$$

The vector $\mathbf{f}^* = [f_1(\mathbf{x}_1^*), \ldots, f_p(\mathbf{x}_p^*)]^T$ is called the Utopia vector. Almost invariably the points $\mathbf{x}_1^*, \ldots, \mathbf{x}_p^*$ are not the same. In other words, there is no $\mathbf{x}^*$ such that $\mathbf{f}^* = \mathbf{f}(\mathbf{x}^*)$. The Pareto front is a convenient tool to choose a suitable compromise between the conflicting objectives.

2. Biobjective optimization algorithm. For simplicity we consider a biobjective problem (1.1) ($p = 2$). In the same way as shown in [20], the algorithm described here extends naturally to constrained problems with $p > 2$.

The algorithm is based on convex combinations of the objectives and homotopy continuation. We introduce the scalar objective function

$$f(\mathbf{x}, \lambda) = (1 - \lambda) f_1(\mathbf{x}) + \lambda f_2(\mathbf{x}), \tag{2.1}$$

where $0 \leq \lambda \leq 1$, and the same problem constraints apply. Standard homotopy starts at $\lambda = 0$ and then steps λ in some fashion, solving the successive subproblems to obtain a discrete sampling of the Pareto set and Pareto front. A frequent and valid criticism of this method is that there is no sure way to obtain a uniform sampling with it because the parametrization of the Pareto front by λ is usually a very nonlinear unknown map.

In the current method we use the old idea of intrinsic parametrization of the Pareto front by using discrete arc length. Let $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ be the minimizers of $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ (obtained by calling a general-purpose optimizer on each single-objective problem (1.2)). Define $\mathbf{x}_0 = \mathbf{x}_1^*$ and $\mathbf{x}_{l+1} = \mathbf{x}_2^*$ to be the end-points of the Pareto set (curve segment) that we construct, where l is the desired number of points in the Pareto front segment joining the end-points.

Given a collection of points $\mathbf{f}_k = \mathbf{f}(\mathbf{x}_k)$, $k = 0, 1, \ldots, l+1$, we define the chord length of a polygon defined by the points as $S_l = \sum_{k=0}^{l} \|\mathbf{f}_{k+1} - \mathbf{f}_k\|$ (where the l_2 norm is used throughout). If the points are samplings of a smooth curve, then when the spacing between the points tends to zero, the chord length tends to the arc length, an intrinsic parametrization of the curve.

Now we define

$$\gamma = \alpha \, \|\mathbf{f}(\mathbf{x}_0) - \mathbf{f}(\mathbf{x}_{l+1})\|/l, \tag{2.2}$$

where the distance between the images of the minimizers times a factor $\alpha > 1$ is an estimate of the total chord length of the Pareto front (accounting for curvature). In the unlikely case that $\mathbf{f}(\mathbf{x}_0) = \mathbf{f}(\mathbf{x}_{l+1})$ we are finished because the Pareto front would be a single point. Otherwise, $\gamma \neq 0$ and we impose the following equispacing constraint:

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}_{\text{prev}}\|^2 = \gamma^2, \tag{2.3}$$

where $\mathbf{f}_{\text{prev}}$ is a previous point in the homotopy process. We then minimize the scalarized function subject to all constraints with λ as an additional variable:

$$\begin{aligned} \min_{\mathbf{x},\lambda} \quad & (1-\lambda) f_1(\mathbf{x}) + \lambda f_2(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}) \leq 0, \quad \mathbf{x} \in D, \quad 0 \leq \lambda \leq 1, \\ & \|\mathbf{f}(\mathbf{x}) - \mathbf{f}_{\text{prev}}\|^2 = \gamma^2. \end{aligned} \tag{2.4}$$

With $\mathbf{f}_{\text{prev}} = \mathbf{f}(\mathbf{x}_0)$, let the solution be $(\mathbf{x}_1, \lambda_1)$. We repeat the process with $\mathbf{f}_{\text{prev}} = \mathbf{f}(\mathbf{x}_1)$ to obtain $(\mathbf{x}_2, \lambda_2)$, and so on. The corresponding discrete Pareto front is defined as the set $\{\mathbf{f}(\mathbf{x}_k),\ k = 0, \ldots, l+1\}$, where $\{\mathbf{x}_k,\ k = 0, \ldots, l+1\}$ is the Pareto set. If the process is successful we obtain a discrete representation of the Pareto front with equispaced values $\mathbf{f}(\mathbf{x}_k)$. Observe that with the use of this intrinsic parametrization, the λ parametrization of the front loses importance.

In order to prevent going backwards to the previous point or straying far away from the current solution, we introduce constraints on the objectives. In case of failure, we provide a restart from the far end going backward, which has proved quite successful.

As initial estimates for the solution of each minimization problem in the homotopy process we use the previous point, plus an estimate of the direction and length of the move (taking into account the constraints, so the estimate remains feasible). No second derivatives of the functionals are required.

Note that constraint (2.3) prevents the subproblems from being convex even if the original goals and constraints are. However, in the convex case, the initial point $\mathbf{x}_0$ is a global minimizer for the single-objective problem (1.2) with $i = 1$. If l is large enough to make γ relatively small, we may argue heuristically that for each problem (2.4) in the homotopy, warm-starting the optimizer at the preceding minimizer is likely to give a *global* minimizer for the current problem. (Recall that SNOPT solves general optimization problems using a sequence of convex QP subproblems.) We see this as a practical benefit of the homotopy approach.

Given our imperfect knowledge of the total arc length, the spacing γ may be inappropriate and we may fall short of or overshoot the target right end-point. A possible way to obtain a tighter result is to do a coarse sampling of the front first and then use the calculated arc length to obtain a better value of γ in (2.2). This is more critical for highly curved fronts.

As we showed in [20], the extension to $p > 2$ objectives is straightforward, although the complexity increases considerably. Since the Pareto set and its image, the Pareto front, have dimension $p-1$, we need that number of weights λ_i, $i = 1, \ldots, p-1$ and also $p-1$ additional distance constraints in order to produce a discrete representation of the Pareto front as a uniform grid of points. Now the parametrization uses chord length along each coordinate axis.

3. A dual formulation. Here we suggest an alternative sequence of subproblems whose *dual* conditions correspond to minimizing the weighted objective function. For variety we describe the general case with p objectives but temporarily omit the constraints. Thus, consider

$$\min_{\mathbf{x}} \ \mathbf{f}(\mathbf{x}), \tag{3.1}$$

where $\mathbf{x} \in R^n$ and $\mathbf{f} \in R^p$. Assume $\mathbf{f}(\mathbf{x})$ is a vector of smooth functions with Jacobian matrix $J(\mathbf{x})$ whose ith row is the gradient of $f_i(\mathbf{x})$. As before, let $\mathbf{x}_i^* = \arg\min_{\mathbf{x}} \ f_i(\mathbf{x})$ and define the symmetric matrix F such that

$$F_{ij} = \|f(\mathbf{x}_i^*) - f(\mathbf{x}_j^*)\|, \qquad F_{\max} = \max_{i,j} F_{ij}, \qquad F_{qr} = \min_{i \neq j} F_{ij},$$

Assuming $F \neq 0$, set $\gamma = F_{\max}/l$ for some integer l such as 10 or 100. With $\mathbf{e}$ being a vector of 1s, set $\mathbf{f}_{\text{prev}} = \mathbf{f}(\mathbf{x}_q^*)$ and consider the optimization problem

$$\min_{\mathbf{x},\,\mathbf{z}} \qquad \mathbf{e}^T\mathbf{z} \tag{3.2}$$

$$\text{subject to} \qquad \mathbf{z} - \mathbf{f}(\mathbf{x}) \geq 0 \quad : \ \mathbf{y} \tag{3.3}$$

$$\|\mathbf{z} - \mathbf{f}_{\text{prev}}\|^2 \geq \gamma^2 \ : \ \mu \tag{3.4}$$

with dual variables $(\mathbf{y}, \mu)$. The optimality conditions for this problem require that

$$J(\mathbf{x})^T\mathbf{y} = 0 \tag{3.5}$$

$$\mathbf{e} - \mathbf{y} - 2\mu(\mathbf{z} - \mathbf{f}_{\text{prev}}) \geq 0 \tag{3.6}$$

$$\mathbf{y} \geq 0, \quad \mu \geq 0. \tag{3.7}$$

We expect at least one of the constraints (3.3) to be active, and constraint (3.4) to be feasible if γ is not too large. We also expect constraint (3.4) to be active. We see from (3.5) that a linear combination of the gradients of each $f_i(\mathbf{x})$ must be zero, and from (3.6)–(3.7) that $\mathbf{y}$ is bounded above and below:

$$0 \leq \mathbf{y} \leq \mathbf{e} - 2\mu(\mathbf{z} - \mathbf{f}_{\text{prev}}).$$

Hence, if the optimal primal and dual variables are $(\mathbf{x}_1, \mathbf{z}_1, \mathbf{y}_1)$, we can say that $\mathbf{y}_1$ is nonnegative and bounded, and that $\mathbf{x}_1$ solves the scalarized optimization problem

$$\min_{\mathbf{x}} \ \mathbf{y}_1^T \mathbf{f}(\mathbf{x})$$

with the elements of $\mathbf{y}_1$ as particular weights on each objective $f_i(\mathbf{x})$.

To trace a path of evenly spaced points on the Pareto front, we solve a sequence of l such optimization problems in which $\mathbf{f}_{\text{prev}}$ in (3.4) is the optimal $\mathbf{f}(\mathbf{x})$ from the previous problem. If the original biobjective problem contains constraints on $\mathbf{x}$, they would be included in the sequence of optimizations.

A drawback of this approach is that the individual objective functions become additional, possibly nonconvex, constraints in the subproblems, and constraint (3.4) is certainly nonconvex. Solvers may therefore be less efficient than in the previous formulation, although the homotopy approach should have a practical advantage as before.

4. Numerical examples. We now apply the method of section 2 to several examples. We use SNOPT [11] to solve the individual nonlinear programs. SNOPT is a general-purpose system for single-objective constrained optimization. It minimizes a linear or nonlinear function subject to bounds on the variables and sparse linear or nonlinear constraints. If gradients of the objective and constraint functions are available, SNOPT uses them; otherwise the unknown gradients are estimated by finite differences. SNOPT uses a sparse sequential quadratic programming (SQP) algorithm. Search directions are obtained from QP problems that minimize a convex quadratic model of the Lagrangian function subject to linearized constraints.

SNOPT requires relatively few evaluations of the problem functions and thus is especially effective if the objective or constraint functions (and their gradients) are expensive to evaluate, as is the case where they are the result of large-scale simulations. (For an interesting example where this approach may be helpful see [1].)

Problem DEB. We start with a biobjective problem from [4] of the form (1.1) with $n = m = 2$:

$$\begin{aligned} f_1 &= x_1, \\ f_2 &= (1 + x_2)/x_1, \\ 9x_1 + x_2 &\geq 6, \\ 9x_1 - x_2 &\geq 1, \end{aligned}$$

with $x_1 \in [0.1, 1]$, $x_2 \in [0, 5]$. The individual objectives have feasible local minima at $\mathbf{x}_1^* = (1/9, 0)$ and $\mathbf{x}_2^* = (1, 0)$.

In Figure 4.1 (left) we see the results of running the homotopy algorithm with $l = 60$. Unfortunately, it seems that the spacing of the points in the Pareto front on the second branch is different from the first—a puzzling result when SNOPT reports very accurate fulfilment of the distance constraint throughout.

The explanation is simple: the aspect ratio in the figure is $10 : 1$ with an amplification that visually magnifies the distance on the almost horizontal second branch. By choosing the scale so that the ratio is closer to $1 : 1$ as in Figure 4.1 (right), we obtain a totally different picture with a much smaller feasible region. The computation took 32 msec on a 3.4 GHz dual processor Athlon PC running Linux (Intel ifort Fortran compiler in debug mode).

Compare with the results from [4] in Figure 4.2, where $50,000$ function evaluations were used (500 generations for a population of 100 points), against about

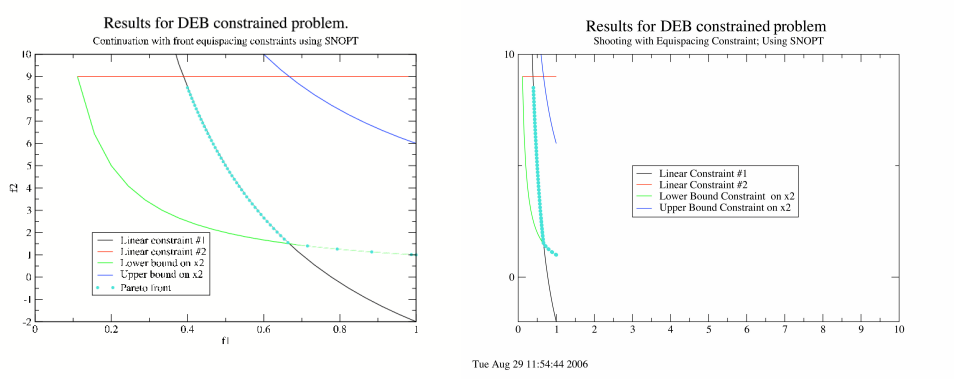

FIGURE 4.1. *Results for problem DEB.*

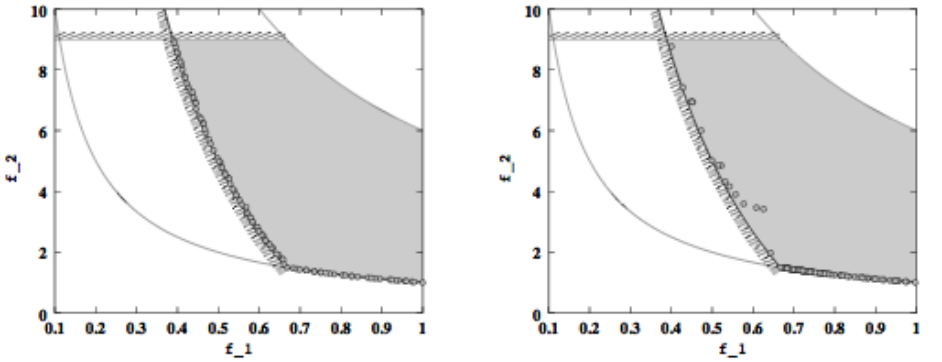

Fig. 15. Obtained non-dominated solutions with NSGA-II on the constrained problem DEB.

Fig. 16. Obtained non-dominated solutions with Ray-Kang-Chye's algorithm on the constrained problem DEB.

FIGURE 4.2. *Problem DEB. Results from [4].*

200 evaluations used by our algorithm (for 60 points). Of course, this result is not surprising, because GAs are notorious for using many function evaluations, making them unsuitable for problems where these function evaluations are costly. Besides, the sampling, although adequate, is hardly uniform.

Problem SRN. The next example is also from [4]:

$$\begin{aligned} f_1 &= (x_1 - 2)^2 + (x_2 - 1)^2 + 2, \\ f_2 &= 9x_1 - (x_2 - 1)^2, \\ x_1^2 + x_2^2 &\leq 225, \\ x_1 - 3x_2 &\leq -10, \end{aligned}$$

with $x_1, x_2 \in [-20, 20]$. We ask for 20 equispaced points in the front. The results are shown in Figure 4.3. The calculation took about 20 msec and required 565 function evaluations (compared to 50,000 for the genetic algorithm in [4]).

Problem NBI. Now we consider an example from [3, 14]. We call it NBI, the name of the method advocated by Das and Dennis in [3], which is shown there as a failure of simple continuation and as a successful application of their method. NBI is

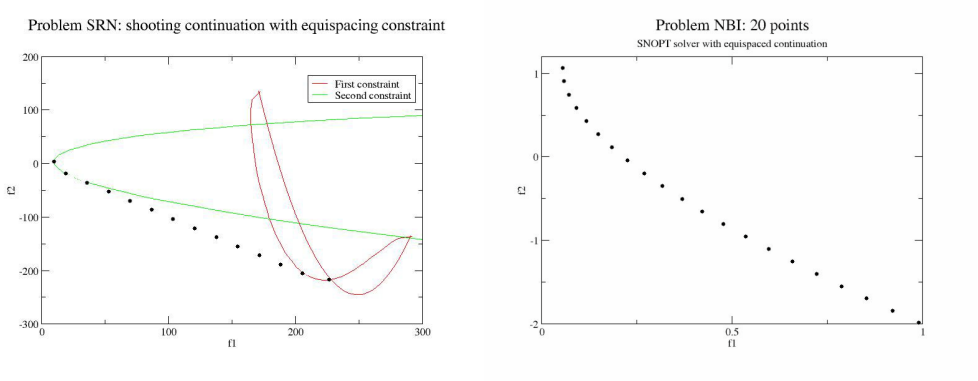

FIGURE 4.3. *Pareto fronts for problems SRN and NBI.*

a biobjective, nonlinear constrained problem with five independent variables:

$$f_1(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2,$$
$$f_2(\mathbf{x}) = 3x_1 + 2x_2 - x_3/3 - 0.01(x_4 - x_5)^3,$$

subject to

$$x_1 + 2x_2 - x_3 - 0.5x_4 + x_5 = 2,$$
$$4x_1 - 2x_2 + 0.8x_3 + 0.6x_4 + 0.5x_5^2 = 0,$$
$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \leq 10.$$

For this example, $l = 20$. In Figure 4.3 we see that our method gives perfectly spaced results (we have scaled the objectives to a maximum value of 1). Our program took 20 msec to generate 21 equispaced points in the Pareto front, running on the same Athlon PC as before. This compares with 3830 msec reported for the adaptive weighted-sum method of [14] and 2430 msec for the NBI method of [3], which also generate well sampled Pareto fronts. (We could not find which computer platform was used in the tests, so these results are not directly comparable to ours, although they were produced in 1998.)

Problem R2a. The previous example showed some escalation in the number of independent variables without any significant effect on the performance of the algorithm. Now we push this up to a problem with 31 independent variables. This problem is a modification (smoothing) of R2, a test problem in [12]. We call it R2a to indicate that is a modified version:

$$f_1(\mathbf{x}) = x_1,$$
$$f_2(\mathbf{x}) = g(\mathbf{x})h(x_1, g(\mathbf{x})),$$

where

$$g(\mathbf{x}) = 1 + 10(n-2) + \sum_{j=2}^{n-1}(x_j^2 - 10\cos(\pi x_j)),$$
$$h(x_1, g(\mathbf{x})) = 1 + e^{-x_1/g(\mathbf{x})} + (x_1 + 1)\sin(\pi x_1)/g(\mathbf{x}),$$

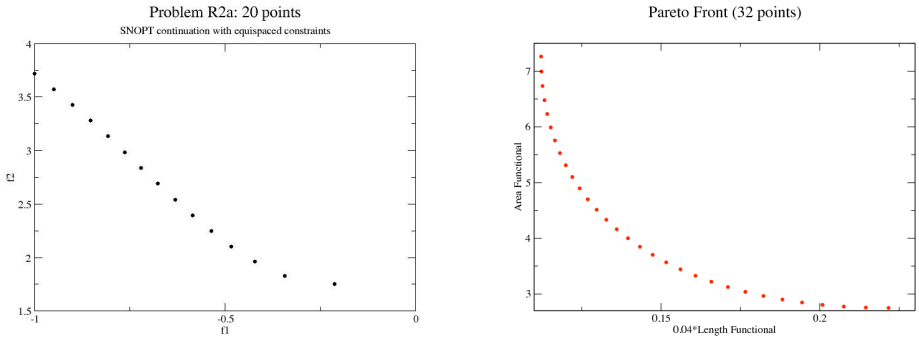

FIGURE 4.4. *Results for problems R2a and Genmesh.*

and $\mathbf{x} \in R^{n-1}$, $-1 \le x_j \le 1$. In Figure 4.4 (left) we show the results for this problem. Again, a perfectly spaced solution of the convex part of the Pareto front is computed in a negligible time.

It is true that these apparently large problems are somewhat artificial: the Pareto set is the interval $-1 \le x_1 \le 1$, with all the other variables identically zero. We understand that the purpose of these made up tests is somewhat different from determining if the algorithms scale well with the number of independent variables. On the other hand, from the point of view of SNOPT, it is proceeding as if the problem were genuinely 31-dimensional. In the next section we show results on an actual application with many variables and non-trivial functionals that would strain evolutionary algorithms.

5. Variational grid generation. In Castillo [2], a method is presented that uses multiple objectives to design "good" meshes for solving partial differential equations in complex domains. As an application of the homotopy method we consider the biobjective optimization problem that uses the grid spacing and grid cell area functionals described in Castillo's paper. An initial mesh is generated from the given geometry and this is optimized using a combination of the criteria. In Castillo's paper the usual scalarization procedure plus trial and error was used to combine the two objectives, leading to better meshes than what could be obtained with any single objective alone. Below, we simply automate that procedure and provide an exploration of the Pareto front to facilitate the choice of the weights.

Problem Genmesh. In order to exercise the homotopy code for generating a discrete Pareto front with equidistant points we have combined Castillo's code with it, including the equispacing constraint and using SNOPT as the optimizer for the scalarized nonlinear programs that result. The unknowns are the Cartesian coordinates of the mesh nodes. In the first example we consider a 2D logically rectangular mesh with 17×17 points.

Only the interior nodes are allowed to vary, resulting in optimization problems with 451 variables (including λ). This problem was originally discussed in [23], where the authors showed how the standard methods of the time (Thompson et al. [26]) produced folded grids. We call this problem Genmesh.

We ran the homotopy continuation with 30 points, obtaining the Pareto front shown in Figure 4.4 (right). In Figure 5.1 we present the meshes that result when one optimizes mesh spacing or area alone. Observe that when mesh spacing is optimized,

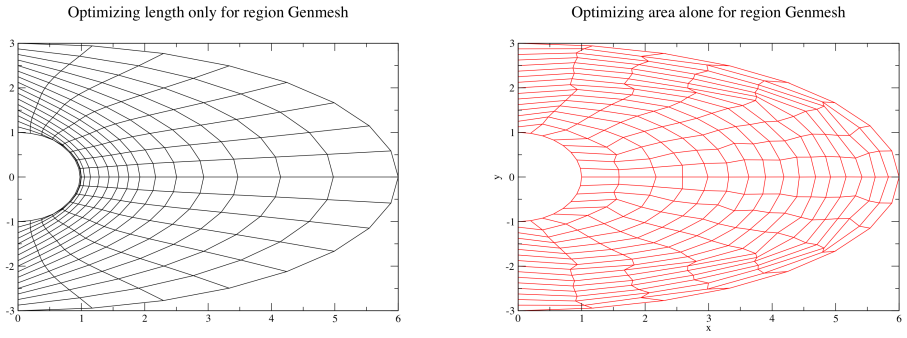

FIGURE 5.1. *Problem Genmesh. Optimizing length and area separately.*

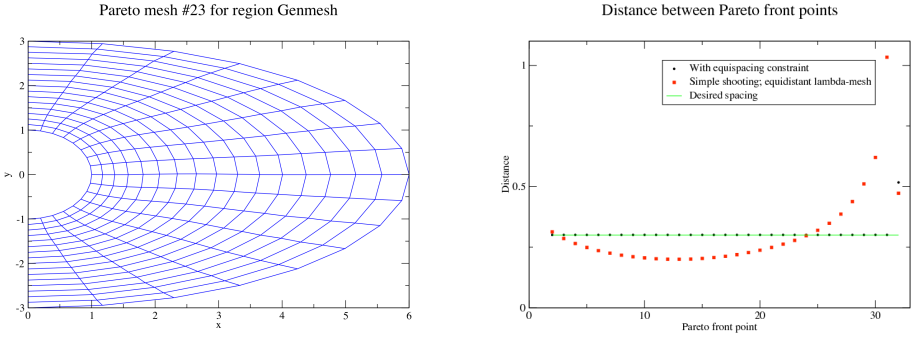

FIGURE 5.2. *Problem Genmesh. Biobjective results.*

mesh lines tends to be close together near the hole, while optimizing area produces a fairly rough mesh. In Figure 5.2 (left) we show the optimal meshes resulting from the combined objectives at the 4th point in the Pareto front (from left to right). As observed in [2], the combined functional gives a better looking mesh than either of the single functionals do. Moreover, when solving an elliptic equation on that domain, one obtains a smaller error for the combined mesh, as we show below.

The time for computing the equispaced Pareto front for this more exigent application was 63 secs on the same platform as above but with the code compiled with the -O flag.

Figure 5.2 (right) illustrates simple continuation with fixed uniformly spaced λ's and no equispacing constraint, as compared to the new algorithm. We have plotted the chord lengths of successive points resulting from both procedures. The full line represents the constant spacing $\gamma = 0.3$. As expected, we see perfect agreement for the points generated by the new procedure (except for the last one where we did not match exactly the total arc length with our estimate), while standard continuation with equispaced λ's has deviations of as much as 300%.

Problem S. The next example corresponds to a region appropriately named "S". The results are shown in Figures 5.3–5.4, including the Pareto front, meshes associated with the end values of λ, and a combined mesh for the 18th Pareto point.

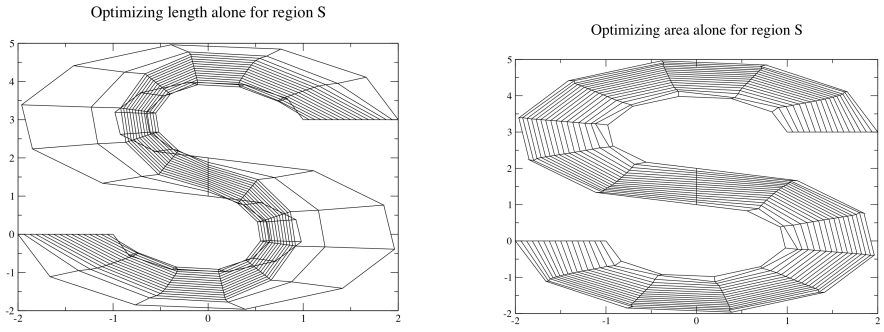

FIGURE 5.3. *Problem S: Optimizing length and area separately.*

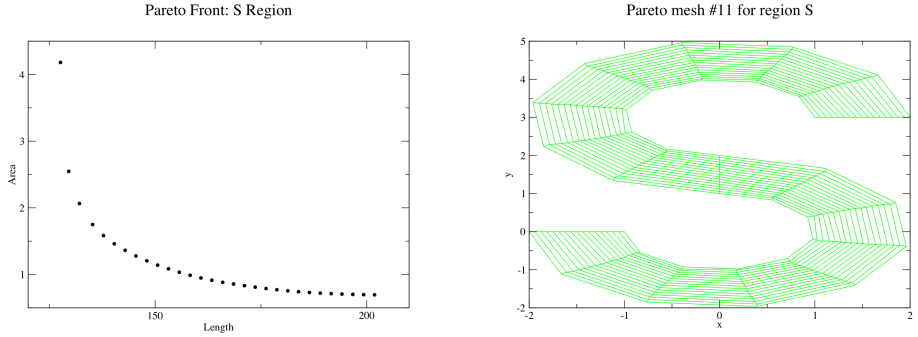

FIGURE 5.4. *Pareto front for region S.*

Of course, good-looking meshes are fine, but what one really wants is to solve differential equations on these meshes with good accuracy. Thus we consider the solution of the following elliptic boundary value problem on the two domains for all of the 17×17 meshes corresponding to the calculated points in the Pareto front:

$$
\begin{aligned}
& L(u) = (\alpha u_x)_x + (\beta u_x)_y + (\beta u_y)_x + (\gamma u_y)_y = g, \\
\text{with} \quad & \alpha = d_1 \cos^2\phi + d_2 \sin^2\phi, && \phi = 5\pi/12, \\
& \beta = (d_1 - d_2)\cos\phi \sin\phi, && d_1 = 1 + 2x^2 + y^2, \\
& \gamma = d_1 \sin^2\phi + d_2 \cos^2\phi, && d_2 = 1 + x^2 + 2y^2.
\end{aligned}
$$

We take the exact solution to be $u_{\text{exact}}(x, y) = \sin \pi x \sin \pi y$, and the Dirichlet boundary conditions are $g(x, y) = L(u_{\text{exact}}(x, y))$ on ∂D. The problem is discretized using a second-order approximation. The Jacobian of the transformation from Cartesian to curvilinear coordinates is calculated approximately by differences. The resulting discretized system is solved using SOR.

In Figure 5.5 we show the maximum error and the Residual Mean Square error for each mesh. We have set the errors arbitrarily equal to 2 if SOR did not converge—a sign of folded or badly distorted meshes that occurs for the meshes associated with the Pareto end-points and also for some of the interior points close to them. The errors tally with the results observed in the Pareto front. For region Genmesh, the errors for points 2–15 are about the same, deteriorating very slowly, while for region

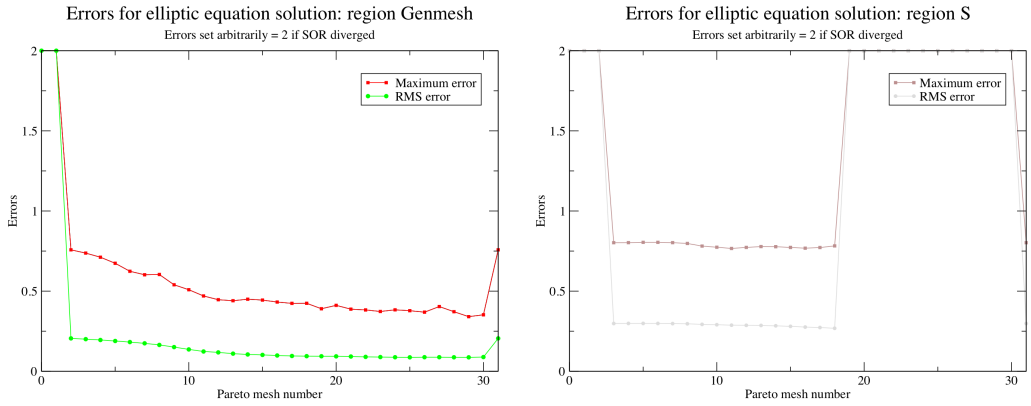

FIGURE 5.5. *Errors for regions Genmesh and S.*

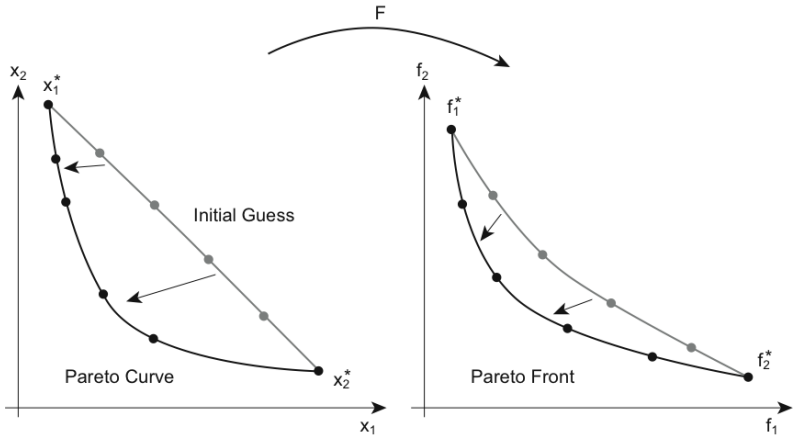

FIGURE 6.1. *Global method.*

S the errors are fairly uniform for meshes associated with points 14–31.

6. Parallel Pareto front calculation. Finally, we describe an algorithm to compute all the points in the Pareto front at once in the manner of Pereyra [20]. This "global" approach contrasts with the "marching" homotopy method described in the preceding sections.

The idea is to pose and solve simultaneously all the scalarized optimization problems that lead to the uniform mapping of the Pareto front; i.e., including the additional equispacing constraints and determining the appropriate weights of the scalarization as part of the process. See Figure 6.1 for a schematic diagram of the process.

This looks like a formidable task because for l points in the Pareto front we would multiply the original number of design variables by l. For the biobjective case, this

leads to a global problem in which all subproblems

$$\begin{aligned} \min_{\mathbf{x}_k,\lambda_k} \quad & (1-\lambda_k) f_1(\mathbf{x}_k) + \lambda_k f_2(\mathbf{x}_k) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}_k) \le 0, \quad \mathbf{x}_k \in D \\ & \|\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_{k-1})\|^2 = \gamma^2 \end{aligned}$$

are required to be solved simultaneously for $k = 1, \ldots, l$. We see that the independent scalarized optimization problems are coupled through the front equispacing constraints, leading to a very sparse and structured global problem, as explained in detail in [20].

Since we are advocating the use of off-the-shelf optimization software such as SNOPT, we pursue now a different idea that is similar in spirit to that of Leyffer [15], in order to achieve an efficient parallel algorithm for large-scale problems (expensive functionals and many variables and constraints). We observe that the marching homotopy continuation of the previous sections is inherently a sequential process and thus is not amenable to parallelization.

The idea is to use an asynchronous iteration [18, 19] in which each individual minimization is assigned to a different processor, while the current values of all the vector functions $\mathbf{f}$ are kept in a central storage that is available to all the processors.

To be precise, the problem solved on processor k is

$$\begin{aligned} \min_{\mathbf{x}_k,\lambda_k} \quad & (1-\lambda_k) f_1(\mathbf{x}_k) + \lambda_k f_2(\mathbf{x}_k) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{x}_k) \le 0, \quad \mathbf{x}_k \in D, \\ & \|\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_{k-1}^{\text{current}})\|^2 = \gamma^2, \end{aligned}$$

where $\mathbf{f}(\mathbf{x}_{k-1}^{\text{current}})$ is obtained from central storage. Each processor updates in central storage its own values of $\mathbf{f}(\mathbf{x}_k)$ as soon as they are calculated. Additional bound constraints can be used to make the homotopy more robust as stated before. Additional bound constraints can be used as indicated before.

This is now an asynchronous block Gauss-Seidel type of iteration. The convergence theory in [18] guarantees robustness of this iteration with regard to potential processor failures.

7. Conclusions. We have presented an efficient homotopy continuation method for solving biobjective optimization problems. The method produces an equispaced sampling of the Pareto front and is very efficient compared with evolutionary algorithms. This will be specially noteworthy for real-life problems, where either the goals or the constraints are expensive to calculate, as we have exemplified with a variational approach to mesh generation for complex 2D regions.

The key concept to obtain an equispaced Pareto front is the addition of a constraint in the homotopy continuation process that explicitly requests that outcome. This simple idea, already advanced in a previous paper of the first author for unconstrained problems [20], seems to be novel and as demonstrated in the numerical examples, it does work as advertised.

For clarity of exposition we have concentrated on the biobjective problem, but the concepts extend naturally to more objectives, although with considerable increase in the cost of implementation, because now one would need to do homotopy continuation in each coordinate direction in order to map the Pareto front, which is an $(m-1)$-dimensional manifold for an m-objectives problem.

Finally, we described a global method that is amenable to parallelization, as required for large-scale problems such as in [1].

Acknowledgements. We are most grateful to Gabriele Eichfelder for her helpful comments.

REFERENCES

[1] J. J. Alonso, P. LeGresley, V. Pereyra: Aircraft design optimization. *Mathematics and Computers in Simulation* 79, 1948–1958 (2009).
[2] J. Castillo: A discrete variational grid generation method. *SIAM J. Sci. Stat. Comput.* 12, 454–468 (1991).
[3] I. Das, J. E. Dennis: Normal-boundary intersection: An alternate approach for generating Pareto-optimal points in multicriteria optimization problems. *SIAM J. Optim.* 8, 631–657 (1998).
[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan: A fast and elitist multi-objective genetic algorithm: NSGA-II. KanGAL Report 2000-01, Indian Institute of Technology, Kanpur (2000).
[5] G. Eichfelder: *Parametergesteuerte Lösung nichtlinearer multikriterieller Optimierungsprobleme* (Parameter-Controlled Solving of Nonlinear Multi-Objective Optimization Problems), PhD dissertation, University of Erlangen-Nurnberg, Germany (2006).
[6] G. Eichfelder: ϵ-constraint method with adaptive parameter control and an application to intensity-modulated radiotherapy. In K.-H. Küfer, H. Rommelfanger, C. Tammer, K. Winkler (eds.), *Multicriteria Decision Making and Fuzzy Systems: Theory, Methods and Applications*, Shaker, Aachen, pp. 25–42 (2006).
[7] G. Eichfelder: *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer (2008).
[8] G. Eichfelder: Personal communication.
[9] G. Eichfelder: A constraint method in nonlinear multi-objective optimization. In V. Barichard et al. (eds.), *Multiobjective Programming and Goal Programming, Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems, Vol. 618 (2009).
[10] G. Eichfelder: An adaptive scalarization method in multi-objective optimization. *SIAM J. Optim.*, to appear (2009).
[11] P. E. Gill, W. Murray, M. A. Saunders: User's guide to SNOPT Version 7: Software for large-scale nonlinear programming. `http://www.scicomp.ucsd.edu/~peg/Software.html` (2006). Accessed 9 October 2008.
[12] A. W. Iorio, X. Li: Rotated test problems for assessing the performance of multiobjective optimization algorithms. *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Seattle, WA*, pp. 683–690 (2006).
[13] H. B. Keller: Global homotopies and Newton methods. *Recent Advances in Numerical Analysis* (Eds. C. de Boor and G. H. Golub), 73–94 (1978).
[14] I. Y. Kim, O. L. de Weck: Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Struct. Multidisc. Optimization* 23, 149–158 (2005).
[15] S. Leyffer: A complementarity constraint formulation of convex multiobjective optimization problems. *INFORMS J. on Computing* 21, 257–267 (2009).
[16] J. M. Ortega and W. C. Rheinboldt: *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York (1970).
[17] A. Pascoletti, P. Serafini: Scalarizing vector optimization problems. *J. Opt. Theory Appl.* 42, 499–524 (1984).
[18] V. Pereyra: Asynchronous distributed solution of large-scale nonlinear inversion problems. In V. Pereyra and J. Castillo (eds.), special issue of *J. Applied Numerical Mathematics* 30, 31–40 (1999).
[19] V. Pereyra: Ray tracing methods for inverse problems. *Inverse Problems*, 16:R1–R35 (2000).
[20] V. Pereyra: Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems. *Mathematics and Computers in Simulation* 79, 1935–1947 (2009).
[21] J. Rakowska, R. T. Haftka, L. T. Watson: Tracing the efficient curve for multi-objective control-structure optimization. *SIAM J. Optim.* 3, 654–667 (1993).
[22] W. C. Rheinboldt, J. Burkardt: PITCON: a program for a locally parametrized continuation process. *ACM TOMS* 9, 236–241 (1983).
[23] P. J. Roache, W. M. Moeny, S. Steinberg: Interactive electric field calculations for lasers. 17th

AIAA Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, CO, 1984. Paper 84-1655 (1984).

[24] S. Sayin. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Math. Prog.* 87, 543–560 (2000).

[25] O. Schutze, A. Dell'Aere, M. Dellnitz: On continuation methods for the numerical treatment of multi-objective optimization problems. *Dagstuhl Seminar Proceedings 04461, Practical Approaches to Multi-Objective Optimization* (2005).

[26] J. F. Thomson, B. K. Soni, N. P. Weatherill (eds.): *Handbook of Grid Generation*, CRC Press (1998).